## REMARKS

Upon entry of this paper, claims 1, 14, and 16 have been amended. In claims 1, 14 and 16 the wording "determining if execution of any fetched instructions will result in a register window overflow condition or a register underflow condition" has been added to further clarify the claimed invention. Further, claims 1 and 16 have been amended to replace the construction "one of A and B", with "or" to preserve the original meaning of the claims. Additionally, claims 14 and 16 have been amended to correct inadvertent errors in wording that could cause confusion. In claim 14 "by" was inserted and in claim 16 "and" was deleted and "by" was inserted. No new matter has been added. Claims 1-19 are currently pending, of which 1, 14 and 16 are independent.

## Claim Rejections under 35 U.S.C. § 102

Claims 1-13 and 16-19 stand rejected under 35 U.S.C. § 102 (e) as being anticipated by U.S. Patent No. 6,631,452 of Lin (hereafter Lin). Claims 14 and 15 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5, 377, 336 of Eickemeyer et al. (hereafter Eickemeyer). Applicants respectfully traverse each of these rejections for reasons discussed below. For ease of the discussion below, each respective related claim set is discussed separately.

Rejection of Claims 1-13 and 16-19 under 35 U.S.C §102:

Claims 1-13 and 16-19 stand rejected under 35 U.S.C. §102 (e) as being anticipated by Lin. Applicants respectfully traverse this rejection in view of the following arguments.

Prior to discussing the substantive rejections below, Applicants wish to provide a brief summary of some of the features relating to what they regard as their invention as claimed in the pending application. This Summary is not intended to convey all of the inventive aspects of the present invention. Instead, this Summary is intended to merely point out some of the features that have been identified as relevant to the rejections stated in the Office Action.

The claimed invention of the pending application is generally directed to a spill/fill engine that detects when a register window spill trap or a register window fill trap is imminent. The spill/fill engine takes steps, responsive to the detector, to avoid the trap that is detected.

In contrast, the invention disclosed in Lin is generally directed to managing data transfers between a backing store and a register stack. The register stack engine monitors activity on the memory channel and transfers data between selected frames of the register stack and a backing store in the memory, responsive to the available bandwidth on the memory channel.

Applicants respectfully submit that the cited prior art reference fails to disclose each and every element of the claimed invention. Applicants submit that Lin fails to disclose, "*a detector for detecting that a register window overflow condition or a register window underflow condition is imminent, by determining if execution of any fetched instructions will result in a register window overflow condition or a register underflow condition,*" as recited in amended claim 1. Likewise, Lin fails to disclose "*determining that a register window overflow condition or a register window underflow condition is imminent by performing a logic operation on a value representative of a state of a register and a value representative of an instruction held in cache and determining if execution of any fetched instructions will result in a register window overflow condition or a register underflow condition,*" as recited in amended claim 16. Lin discloses a register stack engine that transfers data between frames "responsive to the available bandwidth on the memory channel," (Abstract), thus, Lin detects available bandwidth on the memory channel. As was stated in the outstanding Office Action "The system always detects that underflow/overflow is imminent..." The invention of Lin assumes that a register window overflow condition and register window underflow condition will potentially occur at some time in the future, thus it does not detect that a register window overflow condition or a register window underflow condition is imminent by determining if execution of any fetched instructions will result in one of a register window overflow or register window underflow.

The outstanding Office Action states that the system of Lin "detects inactive procedures" and responds. Applicants respectfully submit that detecting inactive procedures is not equivalent to detecting an imminent spill or fill condition, as disclosed in the claimed invention of the present application. Further, the outstanding Office Action cites column 3, lines 59-67, column 4, lines 12-17, and column 4, lines 39-42, as support that Lin discloses a detector for detecting that one of the register window overflow condition and one of a register window underflow condition is imminent. These passages are directed to speculative spills and fills. As was noted

in the previous paragraph, these speculative spills and fills occur in response to available bandwidth and not in response to an imminent underflow/overflow condition. Lin discloses speculative loads and stores to "increase the size of clean partition 430 at the expense of invalid partition," column 7, lines 61-62. In the same section of Lin the speculative loads are more specifically characterized in the patent as "opportunistic load operations" in column 7, lines 61-62. . Lin's criteria for implementing the opportunistic spill operations or fill operations are that no mandatory RSE operations are pending and that there is available bandwidth in the memory channel, column 8, lines 42-47. None of the criteria for implementation of the speculative spills and fills specify an imminent register window overflow condition or register window underflow condition, thus the opportunistic (speculative) operations are not responsive to that condition.

Additionally, Lin fails to disclose, *"an instruction generator responsive to the detector for generating at least one instruction to manipulate the storage to avoid a trap to avoid stalling the microprocessor,"* as recited in claim 1, and, *"in response to determining that the one of the register window overflow condition or register window underflow condition is imminent, manipulating the storage to avoid a trap,"* as recited in claim 16. When faced with an imminent spill or fill, the register stack engine of Lin executes a mandatory register stack engine (RSE) operation, (Figure 5), "If the RSE state indicates that mandatory spill or fill operations are necessary, these are implemented and the RSE state is adjusted accordingly," (column 9, lines 1-3 and Figure 5). The mandatory RSE operation may stall the microprocessor, "Mandatory spill and fill operations may cause the processor to stall if the active procedure can not make forward progress until the mandatory spill/fill operations complete," (column 3, lines 2-5). When faced with an imminent spill or fill Lin uses mandatory operations which may stall the processor, thus, Lin does not disclose all of the elements of claim 1 and claim 16.

Lin does not disclose each and every element of independent claims 1 and 16, which are therefore patentable. Claims 2-13 depend from claim 1 and claims 17-19 depend from claim 16. Accordingly, claims 2-13 and 17-19 are therefore patentable as being dependent on an allowable base claim in addition to their own claimed characteristics. Reconsideration and withdrawal of the rejection of claim 1-13 and 16-19 is requested.

Rejection of Claims 14-15 under 35 U.S.C §102:

Claims 14 and 15 stand rejected under 35 U.S.C. §102(b) as being anticipated by Eickemeyer. Applicants respectfully traverse this rejection in view of the following arguments.

Eickemeyer is generally directed to a load unit for processing the data fetch in load instructions. By processing the fetch early, cache misses can be processed in parallel with other execution thereby reducing the performance degradation of cache misses.

Applicants respectfully submit that the cited prior art reference fails to disclose each and every element of the claimed invention. Applicants submit that Eickemeyer fails to disclose *"a detector for detecting an instruction in a cache prior to execution of said instruction indicating that a trap requiring an access to the storage to manage register window information is imminent, by determining if execution of any fetched instructions will result in a register window overflow condition or a register underflow condition;"* as recited in amended claim 14. Eickemeyer does not disclose a detector as specified in amended claim 14. The previous Office Action cites column 3, lines 30-47 as evidence that Eickemeyer discloses such a detector. Applicants respectfully submit that a detector for load instructions (as disclosed in Eickemeyer) is not equivalent to detector for detecting an instruction in a cache prior to execution of said instruction indicating that a trap requiring an access to the storage to manage register window information is imminent, by determining if execution of any fetched instructions will result in a register window overflow condition or a register underflow condition In Eickemeyer, every load instruction in the instruction buffer results in a prefetch, column 7, lines 1-11. Eickemeyer does not determine if execution of any fetched instructions will result in a register window overflow condition or a register underflow condition.

Further, the detector of Eickemeyer detects load instructions in an instruction buffer, "The load unit 107 detects data fetch instructions from the I-buffer 105, predicts the address of the data to be fetched, and processes the fetch...," (column 7, lines 2-5). This is also illustrated in FIG. 1 where the load unit is in direct communication with the I-BUFFER (instruction buffer). A buffer is not equivalent to a cache. While both are memory, they are different in structure function and operation. A cache is a small fast memory holding recently accessed data, designed to speed up subsequent access to the same data. In contrast a buffer is an area of memory used

for storing messages with an input pointer, and output pointer and a count of the space that is used or free. Replacing a cache with a buffer in a computer, or vice versa, would result in a non-functional computer, thus a buffer is not equivalent to a cache. Eickemeyer does not disclose *"a detector for detecting an instruction in a cache,"* as recited in claim 14.

Eickemeyer does not disclose every element of independent amended claim 14, which is therefore allowable. Claim 15 depends from independent amended claim 14 and, as such, incorporates all the patentable features of claim 15 as well as its own patentable characteristics. Reconsideration and withdrawal of the rejection of claims 14 and 15 is requested.

## CONCLUSION

In view of the above remarks and amendments, applicants respectfully submit that Lin and Eickemeyer fail to disclose each and every element of claims 1-19. Applicants respectfully request the Examiner to reconsider and to withdraw the current rejections and pass the claims into allowance.

Applicants believe no fee is due with this statement. However, if a fee is due, please charge our Deposit Account No. 12-0080, under Order No. SMQ-023RCE2 from which the undersigned is authorized to draw.

Dated: March 23, 2006

Respectfully submitted,

By David R Burns

David R. Burns
Registration No.: 46,590
LAHIVE & COCKFIELD, LLP
28 State Street
Boston, Massachusetts 02109
(617) 227-7400
(617) 742-4214 (Fax)
Attorney/Agent For Applicant